

Lab 13

Strings

GOALS:

1. Students will be able to manipulate text that is inputted from the keyboard.
2. Students will be able to write code involving Strings.

ESSENTIAL QUESTIONS

1. Why is it necessary to manipulate text?
2. How are Strings used today in computers?

Strings Case:

There are several ways to change the case of Strings. Visual Basic has built-in functions, which are predefined meaning that are already created and you can use them in your program. There are 3 built-in functions that change the case of a String.

1. LCase(string)
2. UCase(string)
3. StrConv(string, vbconversion)

LCase(string):

LCase(string) - Converts the string variable to all lowercase. Example:

strWord = "Hello" → strWord = LCase(strWord) → strWord will have "hello" stored in it

UCase(string):

UCase(string) - Converts the string variable to all uppercase. Example:

strWord = "Hello" → strWord = UCase(strWord) → strWord will have "HELLO" stored in it

StrConv(string, vbconversion)

StrConv(string, vbconversion) – Give you 3 ways to change the string. Vbconversion will have to be vbUpperCase, vbLowerCase, or vbProperCase. vbProperCase will make a string first letter capitalized.

Examples: strWord = "hello world"

strWord = StrConv(strWord, vbUpperCase) → strWord will have "HELLO WORLD"

strWord = StrConv(strWord, vbLowerCase) → strWord will have "hello world"

strWord = StrConv(strWord, vbProperCase) → strWord will have "Hello World"

Lesson Strings

In this lesson, we will learn how to use some of the string manipulation function such as Len, Right, Left, Mid, Trim, Ltrim, Rtrim, Ucase, Lcase, Instr, Val, Str, Chr and Asc.

(i)The Len Function

The length function returns an integer value which is the length of a phrase or a sentence, including the empty spaces. The format is

Len (String) → String is a String memory location or String Literal (word in double quotes)

For example,

Len (“VisualBasic”) = 11 and Len (“welcome to VB tutorial”) = 22

(ii) The Right Function

The Right function extracts the right portion of a phrase. The format is

Right (“Phrase”, n)

Where n is the starting position from the right of the phrase where the portion of the phrase is going to be extracted. For example,

Right (“Visual Basic”, 4) = “asic”

(iii) The Left Function

The Left\$ function extract the left portion of a phrase. The format is

Left (“Phrase”, n)

Where n is the starting position from the left of the phase where the portion of the phrase is going to be extracted. For example,

Left (“Visual Basic”, 4) = “Visu”

(iv) The Ltrim Function

The Ltrim function trims the empty spaces of the left portion of the phrase. The format is

Ltrim (“Phrase”)

.For example,

Ltrim (“ Visual Basic”, 4)= “Visual basic”

(v) The Rtrim Function

The Rtrim function trims the empty spaces of the right portion of the phrase. The format is

Rtrim (“Phrase”)

.For example,

Rtrim (“Visual Basic ”, 4) = “Visual basic”

(vi) The Trim function

The Ttrim function trims the empty spaces on both side of the phrase. The format is

Trim (“Phrase”)

.For example,

Trim (“ Visual Basic ”) = “Visual basic”

(viii) The Mid Function

The **Mid** function extracts a substring from the original phrase or string. It takes the following format:

Mid(phrase, position, n)

Where position is the starting position of the phrase from which the extraction process will start and n is the number of characters to be extracted. For example,

Mid(“Visual Basic”, 3, 6) = “ual Bas”

(ix) The Instr function

The **Instr** function looks for a phrase that is embedded within the original phrase and returns the starting position of the embedded phrase. The format is

Instr (n, original phrase, embedded phrase)

Where n is the position where the Instr function will begin to look for the embedded phrase. For example

Instr(1, “Visual Basic”, “ Basic”) = 8

(x) The Ucase and the Lcase functions

The **Ucase** function converts all the characters of a string to capital letters. On the other hand, the **Lcase** function converts all the characters of a string to small letters. For example,

Ucase(“Visual Basic”) = “VISUAL BASIC”

Lcase(“Visual Basic”) = “visual basic”

(xi) The Str and Val functions

The **Str** is the function that converts a number to a string while the **Val** function converts a string to a number. The two functions are important when we need to perform mathematical operations.

(xii) The Chr and the Asc functions

The **Chr** function returns the string that corresponds to an ASCII code while the **Asc** function converts an ASCII character or symbol to the corresponding ASCII code. ASCII stands for “American Standard Code for Information Interchange”. Altogether there are 255 ASCII codes and as many ASCII characters. Some of the characters may not be displayed as they may represent some actions such as the pressing of a key or produce a beep sound. The format of the Chr function is

Chr(charcode)

and the format of the Asc function is

Asc(Character)

The following are some examples: Use the ASCII chart to find numbers

Chr(65)=A

Chr(122)=z

Chr(37)=%

Asc(“B”)=66

Asc(“&”)=38

PROJECT 1

1. Code the following below:

Option Explicit

```
Private Sub optUpper_Click()
```

```
    Dim strPhrase As String
```

```
    strPhrase = txtString.Text
```

```
    lblConvertedString = UCase(strPhrase) ' Displays string as uppercase
```

```
End Sub
```

```
Private Sub optLower_Click()
```

```
    Dim strPhrase As String
```

```
    strPhrase = txtString.Text
```

```
    lblConvertedString = LCase(strPhrase) ' Displays string as lowercase
```

```
End Sub
```

```
Private Sub optProper_Click()
```

```
    Dim strPhrase As String
```

```
    strPhrase = txtString.Text
```

```
    lblConvertedString = StrConv(strPhrase, vbProperCase) ' Displays string as initial uppercase letters
```

```
End Sub
```

QUESTIONS

1. What do the following functions do?
 - a. UCase(string) –
 - b. LCase(string) –
 - c. StrConv(string, vbProperCase) –
 - d. StrConv(string, vbUpperCase) –
 - e. StrConv(string, vbLowerCase) –

Project 2

Manipulating Strings

There are 3 built-in functions that allow us to manipulate strings.

1. Left(string, #)
2. Right(string, #)
3. Mid(string, #, #)

We also have a built in functions that allows to find the length of a String called Len(string), which returns the number of letters of the String.

Strings have an index for **each** letter.

Example “Sunshine”

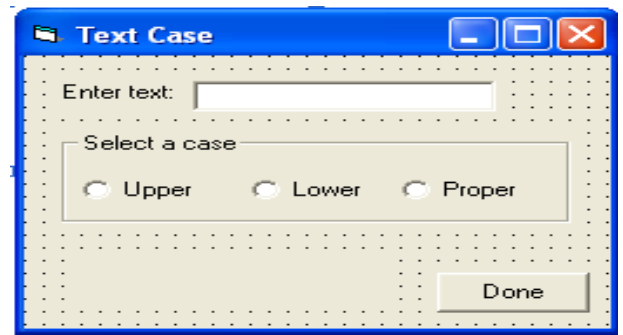
S u n s h i n e

1 2 3 4 5 6 7 8

Examples:

Let strWord = “Sunshine”

strHold = Left(strWord, 3) → strHold will have “Sun” in it.



strHold = Right(strWord, 3) → strHold will have “ine” in it.

strHold = Mid(strWord, 5,1) → strHold will have “h” in it since h is in the 5th position and we take 1 letter.

PROJECT 2

1. Code the following:

Option Explicit

```
Private Sub txtString_Change()  
    lblFirstLetter.Caption = ""  
    lblMiddleLetter.Caption = ""  
    lblLastLetter.Caption = ""  
End Sub
```

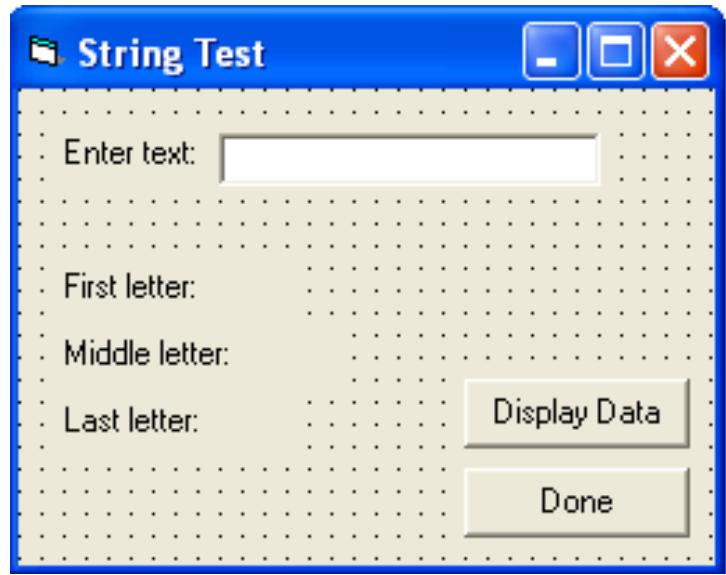
```
Private Sub cmdDisplayData_Click()  
    Dim strPhrase As String  
    Dim intMiddleLetter As Integer
```

```
    strPhrase = txtString.Text  
    lblFirstLetter.Caption = Left(strPhrase, 1)
```

```
    If Len(strPhrase) Mod 2 <> 0 Then                ' Determine middle letter  
        intMiddleLetter = Len(strPhrase) \ 2 + 1  
    Else  
        intMiddleLetter = Len(strPhrase) \ 2  
    End If
```

```
    lblMiddleLetter.Caption = Mid(strPhrase, intMiddleLetter, 1)  
    lblLastLetter.Caption = Right(strPhrase, 1)
```

```
End Sub
```



QUESTIONS

1. What is the purpose of the following code? Explain

```
If Len(strPhrase) Mod 2 <> 0 Then  
    intMiddleLetter = Len(strPhrase) \ 2 + 1  
Else  
    intMiddleLetter = Len(strPhrase) \ 2  
End If
```

2. What is the purpose of the following code? Explain

```
lblMiddleLetter.Caption = Mid(strPhrase, intMiddleLetter, 1)
```

PROJECT 3

InStr(#, string1 , string2) returns the position of the string that you are hunting for. If it can't find the string it returns 0.

Example:

```
strWord = "sunshine"
```

intnum = InStr(1, strWord, "shine") → intnum will get 4 because that is the position where “shine” starts.

Project 3

1. Code the program below

Option Explicit

```
Private Sub txtString_Change()
```

```
    lblMessage.Caption = ""
```

```
    lblPosition.Caption = ""
```

```
End Sub
```

```
Private Sub txtSearchString_Change()
```

```
    lblMessage.Caption = ""
```

```
    lblPosition.Caption = ""
```

```
End Sub
```

```
Private Sub cmdFindString_Click()
```

```
    Dim strPhrase As String, strSearchPhrase As String
```

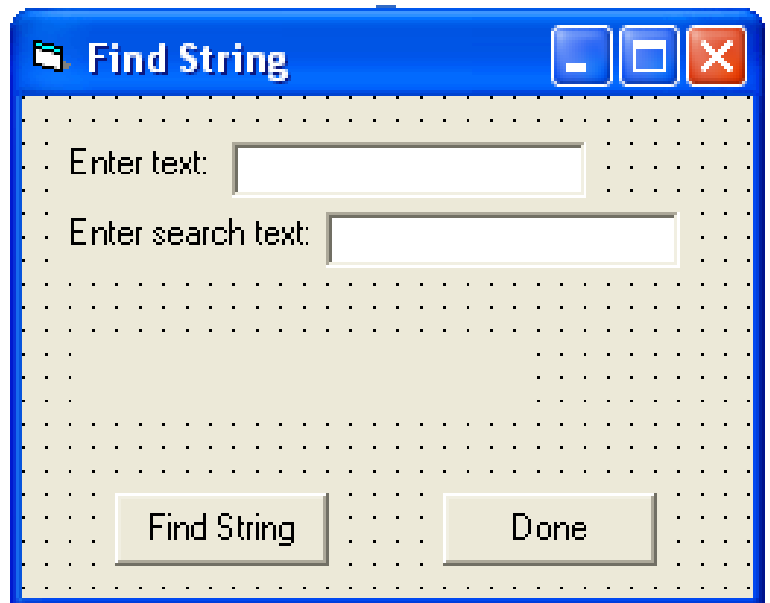
```
    strPhrase = txtString.Text
```

```
    strSearchPhrase = txtSearchString.Text
```

```
    lblMessage.Caption = "Position of search text:"
```

```
    lblPosition.Caption = InStr(strPhrase, strSearchPhrase)
```

```
End Sub
```



Questions

1. What is returned if the InStr() function cannot find the search phrase of the String? Try it?
2. Is it Case Sensitive, meaning if you are search for a substring and it is of a different Case (lower or upper), will the InStr() Function find it? Try it:

Project 4

Len() FUNCTION

The Len() function returns the number of characters a String contains. Example:

```
strWord = "Sunshine"
```

```
intX = Len(strWord) 'intX will receive 8
```

1. Code the program below

Option Explicit

```
Private Sub txtLetter_Change()
```

```
    lblMessage.Caption = ""
```

```
    lblNumberOfOccurrences.Caption = ""
```

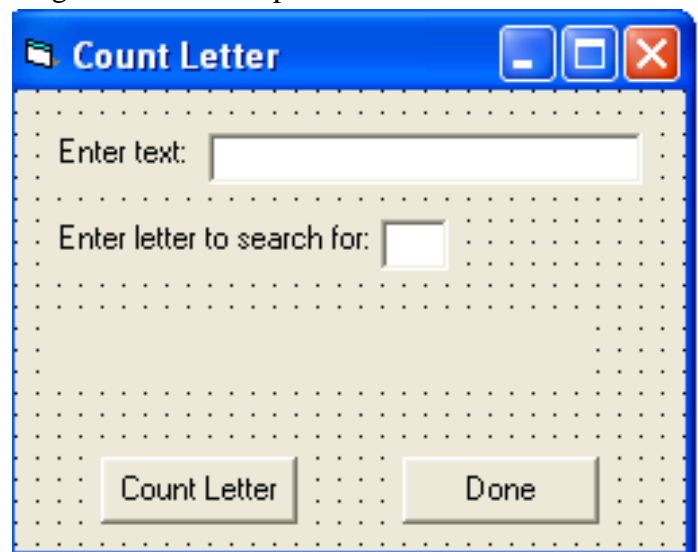
```
End Sub
```

```
Private Sub txtString_Change()
```

```
    lblMessage.Caption = ""
```

```
    lblNumberOfOccurrences.Caption = ""
```

```
End Sub
```



Private Sub cmdCountLetter_Click()

Dim intNumberOfOccurrences As Integer, intSearchPosition As Integer, intLetterPosition As Integer
Dim strPhrase As String, strLetter As String

intNumberOfOccurrences = 0

intSearchPosition = 1

strPhrase = txtString.Text

strLetter = txtLetter.Text

Do

intLetterPosition = InStr(intSearchPosition, strPhrase, strLetter)

If intLetterPosition > 0 Then

intNumberOfOccurrences = intNumberOfOccurrences + 1

intSearchPosition = intLetterPosition + 1

End If

Loop While intSearchPosition <= Len(strPhrase) And intLetterPosition <> 0

lblMessage.Caption = "Number of times the letter occurs:"

lblNumberOfOccurrences.Caption = intNumberOfOccurrences

End Sub

Questions

1. What did the Len() Function return?
2. How does the Len() function control the loop?
3. Why did you need to include **intLetterPosition <> 0** in the loop?

PART 4B

SPACE() FUNCTION

The Space() functions inputs the number of spaces that's in the argument --> (). Example:

lblDisplay.Caption = "Hello" & Space(8) & "World" → ' Output is Hello World

1. Program the following

Option Explicit

Private Sub Form_Load()

Dim strFirstName As String, strLastName As String, strFullName As String

Dim intLengthOfString As Integer

Dim strBar As String

strFirstName = InputBox("Enter your first name:", "First Name")

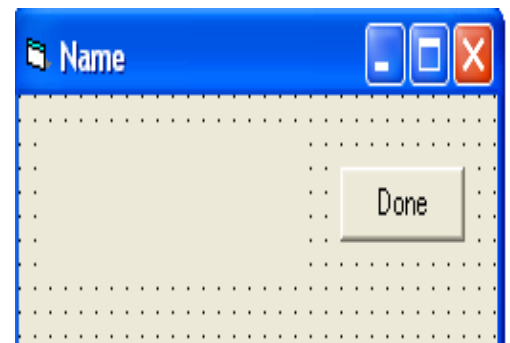
strLastName = InputBox("Enter your last name:", "Last Name")

strFullName = strFirstName & Space(1) & strLastName

lblName.Caption = strFullName

intLengthOfString = Len(strFullName)

strBar = String(intLengthOfString, "_")



```
lblStars.Caption = strBar
End Sub
```

QUESTIONS

1. How many spaces were inserted between the first and last name?
2. What is the purpose of the following piece of code? Explain;
strBar = String(intLengthOfString, "_")

Project 5:

StrComp() FUNCTION

The StrComp() function allows the user to compare strings. Example:

StrComp(string1, string2, compare) → returns 0 if string1 and string2 are equal, returns 1 if string1 is greater than string2, and -1 if string1 is less than string2. Use if statement and compare to 0 to determine >, <, or =.

1. Code program below

Option Explicit

```
Private Sub txtWord1_Change()
    lblMessage.Caption = ""
End Sub
```

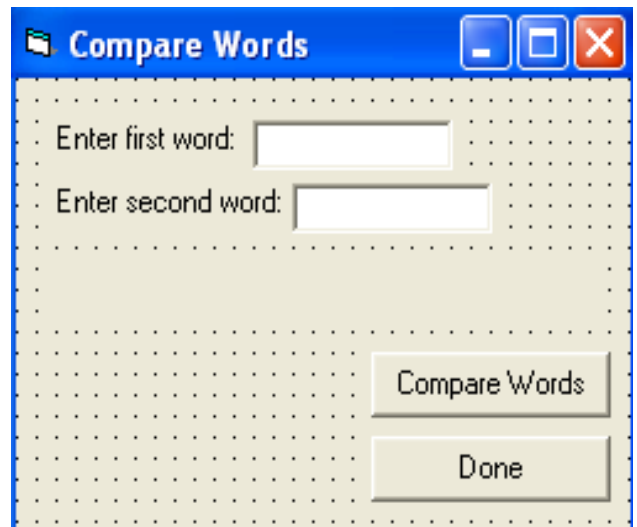
```
Private Sub txtWord2_Change()
    lblMessage.Caption = ""
End Sub
```

```
Private Sub cmdCompareWords_Click()
    Dim strFirstWord As String, strSecondWord As String
    strFirstWord = txtWord1.Text
    strSecondWord = txtWord2.Text
```

```
    If StrComp(strFirstWord, strSecondWord, vbTextCompare) = 0 Then
        lblMessage.Caption = strFirstWord & " is the same as " & strSecondWord
    ElseIf StrComp(strFirstWord, strSecondWord, vbTextCompare) = -1 Then
        lblMessage.Caption = strFirstWord & " comes before " & strSecondWord
    Else
        lblMessage.Caption = strFirstWord & " comes after " & strSecondWord
    End If
End Sub
```

Questions

1. What are the three possible outcomes of the strComp() Function?
2. What is the purpose of the argument vbTextCompare in the StrComp() function?



PROJECT 6

1. Code the program below

Option Explicit

```
Private Sub cmdPlayGame_Click()
```

```
    Const strSentinel As String = "!"
```

```
    Dim strSecretWord As String, intSecretWordLength As Integer
```

```
    Dim intNumberOfGuesses As Integer
```

```
    Dim strGuess As String, strWordGuessedSoFar As String
```

```
    Dim intLetterPos As Integer
```

```
    Dim intScore As Integer
```

```
    Dim blnGuessCorrect As Boolean
```

```
    strSecretWord = "magic"
```

```
    intSecretWordLength = Len(strSecretWord)
```

```
    strWordGuessedSoFar = String(intSecretWordLength, "-")
```

```
    lblWord.Caption = strWordGuessedSoFar
```

```
    intNumberOfGuesses = 0
```

```
    intScore = 100
```

```
    lblMessage.Caption = "Number of guesses = " & intNumberOfGuesses & vbCrLf & "Score = " &  
intScore
```

```
    strGuess = InputBox("Guess a letter (! to guess word)", "Hangman")
```

```
    Do While strGuess <> strSentinel
```

```
        intNumberOfGuesses = intNumberOfGuesses + 1
```

```
        For intLetterPos = 1 To intSecretWordLength
```

```
            If StrComp(strGuess, Mid(strSecretWord, intLetterPos, 1), vbTextCompare) = 0 Then
```

```
                Mid(strWordGuessedSoFar, intLetterPos, 1) = strGuess
```

```
                blnGuessCorrect = True
```

```
            End If
```

```
        Next intLetterPos
```

```
        If Not blnGuessCorrect Then
```

```
            intScore = intScore - 10
```

```
        End If
```

```
        blnGuessCorrect = False
```

```
        lblWord.Caption = strWordGuessedSoFar
```

```
        lblMessage.Caption = "Number of guesses = " & intNumberOfGuesses & vbCrLf & "Score = " &  
intScore
```

```
        strGuess = InputBox("Guess a letter (! to guess word)", "Hangman")
```

```
    Loop
```

```
    If strGuess = strSentinel Then
```

```
        strGuess = InputBox("Guess the word")
```

```
    End If
```

```
    If StrComp(strGuess, strSecretWord, vbTextCompare) = 0 Then
```

```
        MsgBox "You win! It took you " & intNumberOfGuesses & " guesses."
```

```
    Else
```



```
    MsgBox "You lose. Press OK to display secret word."  
End If
```

```
    lblWord.Caption = strSecretWord  
End Sub
```

Enjoy the Game! ☺